SAGA: tasks & task bulks

Realization of bulk operations within the C++ reference implementation

at GGF17, Tokyo, Japan,
Stephan Hirmer

Center for Computation & Technology
AT LOUISIANA STATE UNIVERSITY

AGENDA

- Introduction: the SAGA task model
- The big picture
- Adding meta-information to saga::task
- Analyzing & bundling task bulks
- Executing bulks
- Monitoring tasks – monitoring bulks
- Demo
- Results

The SAGA task model

- saga::task
  - saga::task::run()
  - saga::task::wait(double timeout)
  - saga::task::cancel()
  - saga::task::get_state()
- saga::task_container
  - saga::task_container::run()
  - saga::task_container::wait()
  - saga::task_container::cancel()
  - saga::task_container::add_task(saga::task)

The big picture

add meta-information to the tasks

bundle tasks into bulks by analyzing them

execute “similar” task together (bulks)

monitor tasks & task bulks

```
ath);
    saga::t
                                    rc,dest[i],…)
    tc.add(t
}
tc.run();
[…] // some other code
tc.wait(-1.0);
[…] // continue …
```

Adding meta-information

- recently:
  - bound functor was put in a task
  - functor was executed within a thread
- now:
  - functor separated from parameters
    + meta-information are put in the task
  - functor is executed with parameters in a thread

⇒task controls parameters + meta-info

Analyzing & bundling tasks

- in task_container::run()
  - using meta-information for analysis
  - bundling “similar” tasks together
- according to different strategies
  - same operation
  - same api object
  - etc …

Executing bulks (I)

- Two different strategies:
  - common points
    - foreach bundle of “similar” tasks
      - selecting suitable bulk-adaptor
      - organizing parameter passing from task to selected bulk-adaptor by using a visitor concept
  - simple strategy
    - assuming that the adaptor is able to execute all the tasks within the bulk

Executing bulks (II)

- more improved strategy
  - only trying to execute a given sub-bulk shows if a selected bulk-adaptor is really able to execute the given sub-bulk
  - selected bulk-adaptor tries to execute all the tasks using its specialized bulk handling
  - returns a subset (may be empty) of tasks he couldn't execute
  - new bulk-adaptors are selected until all bulks are executed
  - if necessary, fall back to one-by-one execution.

Monitoring bulks

- Problem:

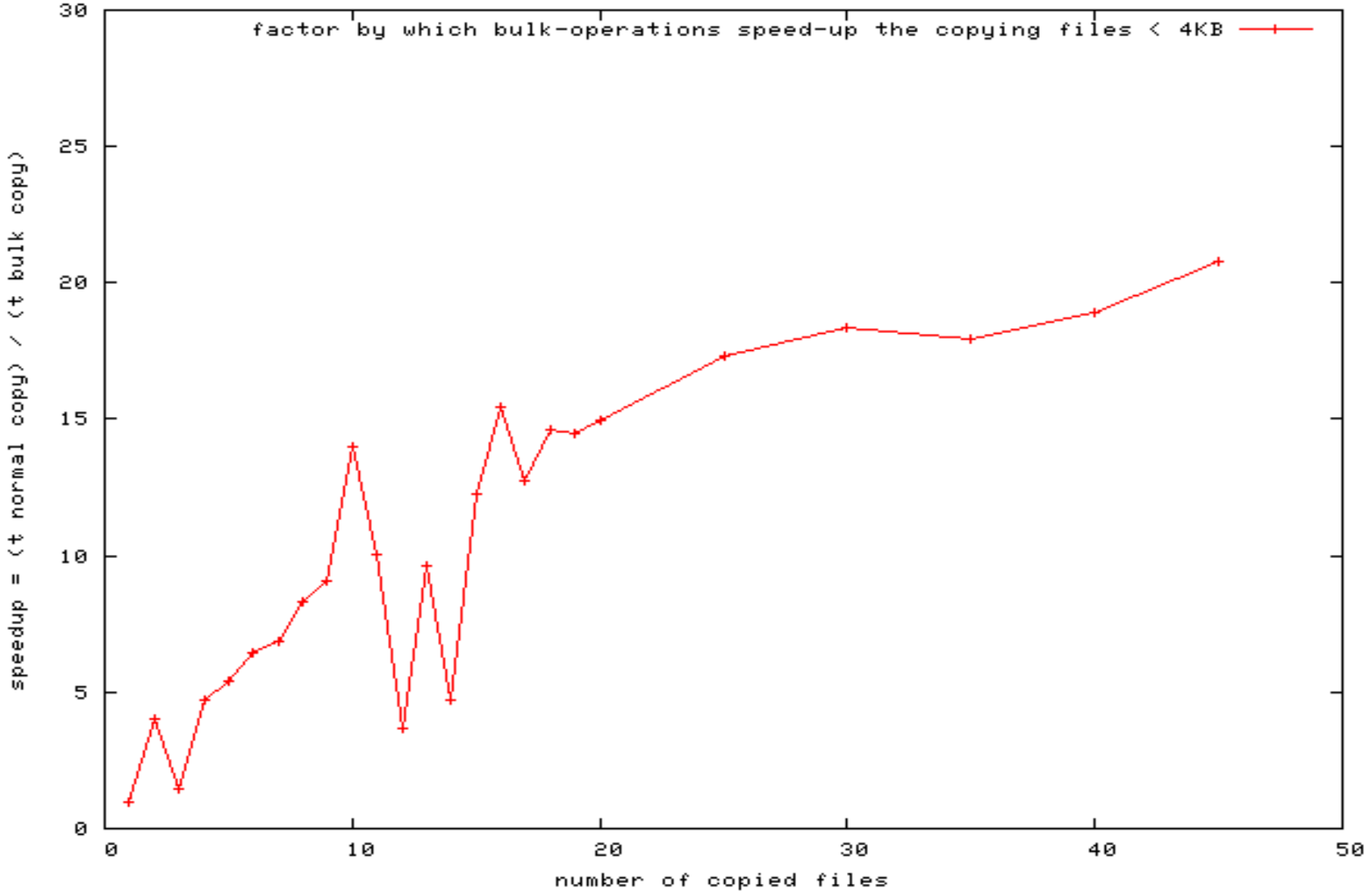
  During the execution of a task within a bulk, the link between the task-object & the real execution gets lost.

- Solution:

  maintain list of links between task-object and the executing bulk-adaptor.

  – using uuids

Results (I) - Benchmarks



Results (II)

- Problems:
  - SAGA specific
    - saga::task_container not derived from saga::task
    - no templates allowed in api-classes
  - General:
    - globus not available for Windows

thank you very much for your attention …